APPLICATION NO. 09/823,331 **DOCKET NO. P-24,723 USA**

REMARKS

In view of the following remarks responsive to the Office Action dated February 25, 2005, Applicant respectfully requests favorable reconsideration of this application.

The Office has rejected all claims under 35 U.S.C. §102 (e) as being anticipated by Anders. Applicant respectfully traverses.

The Present Invention

The present invention is a method and apparatus for loading web pages, including supplemental files such as pictures, sound files, video files, etc., at a browser. One of the problems of the prior art addressed by the present invention is that browsers typically read the HTML code in a Web page from left to right and from top to bottom. Accordingly, the browser encounters the embedded references to such supplemental files in the order in which they are encountered while reading the page. The browser will send requests back to the server for those supplemental files in the order that the browser encounters the references while reading the HTML code. Since a browser has a limited number of ports, the supplemental files may not be retrieved and loaded in the most efficient manner. For instance, if a browser has four ports and the requested page has 14 supplemental files, in which the first four referenced supplemental files are large files and the next 10 are small files, the browser may take a long time to download the first four large files, while the person sitting at the client browser watches a largely or completely blank screen. If the ten small files could be

downloaded first, the browsing experience for the person can be much improved because he/she could then have something to look at while waiting for the four large files to download.

The present invention addresses this concern without the need to modify the browser software in any way. In accordance with the invention, the order in which supplemental files referenced in a Web page are downloaded from the server to the requesting client is specified by the designer of the HTML code of the Web page and controlled at the server side regardless of the order in which the client-side Web browser encounters and requests the supplemental files. Particularly, each supplemental file referenced in a Web page has a sequence number associated with it. In a preferred embodiment, the sequence number is provided as an additional attribute of the tag that calls the supplemental file. Since the client Web browser is a standard Web browser, it will have no idea what the sequence attribute is, which is irrelevant because a browser will simply ignore any attribute within a tag that it does not understand. However, at the server-side, when the page is requested, the server parses the page before sending it to the requesting client to find the tags for the supplemental files embedded within the page and reads the associated sequence number attributes. It then builds a queue for serving the supplemental files to the client machine, the supplemental files being queued in the order dictated by the sequence numbers.

Thereafter, regardless of the order in which the browser returns requests for the supplemental files, the server will serve the supplemental files in the order dictated by the queue. Existing browsers already are equipped to receive and cache files and associate such cached files with files referenced in an HTML page. Accordingly, the fact that the supplemental files referenced in a Web page may be received in an order different from the order in which the browser requests them is of no consequence. Accordingly, the invention resides entirely at the server side and will work with any Web browser.

The Anders Reference

Anders discloses a method and apparatus for serving Web pages, including supplemental files, to a requesting client. However, the method and apparatus disclosed in Anders is entirely different from that of the present invention. Most notably, unlike the present invention, Anders requires the Web browser software to be modified to function with the invention. See col. 8, lines 2-6 (indicating the need for a Jammer unpacker "on the client"), col. 8, lines 51-54, and col. 12, line 65 – col. 14, line 7 (which describes in detail the software needed at the browser to implement the invention).

Anders' scheme is entirely different than Applicant's. With reference to Anders' Figure 8, the server transmits the requested page to the requesting client in a particular data stream format 190 that includes the data for the main object (the Web page) and the data for the supplemental objects (such as embedded pictures, etc.) in data entries (packets, such as packets 181-189) that are interleaved with each other in an order

selected by the developer. More particularly, the data stream 190 comprises a stream header 180 at the beginning of the stream followed by data definition entries and HTML data entries. Each data definition entry, e.g., 181, 182, 185, 187, defines a supplemental object/file present in the Web page data stream. There is one data definition entry per object/file. The HTML data entries are the actual data of the objects/files (including the main file as well as the supplemental files). Each file will typically consist of many HTML data entries that the browser assembles together to render the whole file. The data definition entry that defines any given object/file must precede the first HTML data entry of that file in order for the browser to know what to do with those HTML data entries when it receives them.

The basic premise of Anders' invention is that the publisher 210 (Fig. 11) interleaves the data for the entire web page in a way dictated by itself and serves it to the client that way. The browser, upon receiving each data definition entry, creates an entry in an unpacked object cache (UOC). Then, when the browser starts receiving the HTML data entries corresponding to the supplemental file identified by any given data definition entry, it will append that HTML data to the entry it created in its UOC. In Anders, the browser receives the tags identifying the supplemental files in the order dictated by the data stream 190. Accordingly, the browser may be receiving data of a supplemental file before it receives the HTML data entry that contains the reference to that supplemental file.. That is not a problem. Particularly, when the browser reaches the reference to the supplemental file that it has already started downloading and

caching in its UOC, the UOC simply forwards the cached data to the browser for rendering.

Discussion

While Anders discloses an interesting technique, it is entirely different from the present invention. In Anders, there is nothing that resembles the sequence number attribute embedded within the tag referencing the supplemental file. Furthermore, the server does not parse the code being sent to the client to detect the sequence numbers. There are no sequence numbers. Rather, Anders' server builds the data stream 190 using a software module that Anders calls the publisher 210 (see figure 11). In Anders, the user specifies the order in which supplemental objects/files are downloaded at the browser, but the information dictating the order is not embedded within the main Web page itself. Rather, the order is determined by an external software module, namely, the Stream Configurator. Thus, while Anders' technology does permit the server to dictate the order in which supplemental files are delivered to the browser, it does so in a way that is entirely different from what is claimed in the present application.

Referring to claim 1, Anders does not disclose (1) "parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served".

Applicant has already presented this argument in response to the last Office Action, in which the same rejection was asserted. The Office replied, arguing:

Applicants argue that Anders does not disclose, "parsing the code comprising the requested page to detect data within the code that indicates in order in which said supplemental files are to be served". In response to Applicant's argument, the Patent Office maintain the rejection because Anders does teach parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served as shown in Col. 10, lines 1-16, 37-40, and 52-62. Anders clearly shows streaming configurator parses web page to identify references to objects and their locations within a page, the designer supplies display sequence information.

The above quoted portion of the Office Action itself discloses the error in the Office's analysis of this issue. The Office is absolutely correct in its statements that Anders teaches that the Streaming Configurator parses the Web page to identify references to objects and their locations within the page, but that it is the designer that supplies the display sequence information. This is the opposite of what is claimed in claim 1, i.e., that the sequence for retrieving the supplemental files is given by data embedded within the Web page itself. As noted above, in Anders, it is the Streaming Configurator software module that dictates the order (as specified by the user) in which the supplemental files will be downloaded; it is not HTML tags or any other form of data within the web page that dictates the order.

Claims 2-8 depend from claim 1 and, therefore, distinguish over Anders for at least all of the reasons given above in connection with claim 1.

However, in addition, the dependent claims add even further distinguishing features. For instance, claim 7 depends from claims 1 and 2 and adds that the references to the supplemental files "comprise HTML tags, and said order data comprises attributes of said tags". There is nothing in Anders remotely resembling this

since the order is dictated by the Streaming Configurator. The Office asserted that this is disclosed in Col. 11, line 7-col. 12, line 44. However, it is quite clear from col. 11, lines 44-47 that the data identifying the display order does not come from HTML tags within the file. Col. 11, lines 44-47 state "User supplied display sequence information. This provides information to the Interleavor for the order in which to display the objects".

Claim 8 depends from claim 7 and further adds that "said order data attributes are not recognizable by said client machine". This is directly contrary to Anders, in which the client machine must be modified in accordance with Anders' technology in order to recognize Anders' data stream.

Independent claim 9 also distinguishes over Anders. Claim 9 includes the limitation of "second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file". Hence, claim 9 recites similar features as claims 1 and 7, but in language of differing scope. Therefore, claim 9 distinguishes over Anders for at least all of the reasons discussed above in connection with claims 1 and 7.

Applicant has already argued in response to the last Office Action, in which the same rejection was asserted, that Anders does not teach this feature. The Office replied, arguing:

> Applicants argue that Anders does not teach, "a second code associated with each of references and comprising an attribute of the tag associated with supplemental file". In response to Applicant's argument, the Patent Office maintain the rejection because Anders does teach a second code associated with each of references and comprising an attribute of a tag associated with

supplemental file as shown in Col. 11,-Col. 12 line 44. Anders clearly shows that a second code associated with each of references and comprising an attribute of a tag associated with supplemental file.

However, as just noted, the "second code" in Anders exists in the streaming configurator, and is not an attribute of a tag.

Claim 11 depends from claim 9 and, therefore, distinguish over Anders for at least all of the same reasons as independent claim 11. In addition, claim 11 further distinguishes over Anders by further describing that the tag comprising the sequence number is an HTML tag. Anders, which does not have a sequence number tag at all, obviously cannot teach such limitations.

Independent claim 12 also distinguishes over Anders by virtue of reciting "program code for parsing said code defining the Web page to detect said order data". Since the sequence information is not in the Web page in Anders, it obviously cannot retrieve that information by parsing the code of the Web page.

Applicant previously made this argument in response to the previous Office Action. In reply thereto, the Office asserted:

> Applicants argue that Anders does not teach "program code for parsing code defining the web page to detect order data and constructing a queue in a memory comprising a list of supplemental files in order". In response to Applicant's argument, the Patent Office maintain the rejection because Anders does teach a program code for parsing code defining the web page to detect order data and constructing a queue in a memory comprising a list of supplemental files in order as shown in col. 11, lines 7-30. Anders clearly shows that a program code for parsing code defining the web page to detect order data and constructing a queue in a memory comprising a list of supplemental files.

However, as described above, Anders does not obtain the order data from inside the web page. Furthermore, the order is given by the manner in which the data is sequenced in the packet. There is no list of the file order. The data itself is simply ordered within the packet in the order desired by the user. This is an entirely different concept. Therefore, it also does not meet the limitation of "constructing a queue in a memory comprising a list of supplemental files in order".

In addition, dependent claim 17 adds "said references to supplemental files comprise HTML tags" and that "said order data comprises attributes of said tags". As discussed above in connection with claim 7, these limitations are not found in Anders. Claim 18 depends from claim 17 and further adds that "said order data attributes are not recognizable by said client machine". This is not found in Anders as discussed above in connection with claim 8.

Independent claim 19 includes the limitations "code for parsing said code defining a Web page to detect said order data", "code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order", and "code for serving said supplemental files to said requesting client machine in said order of said queue". In Anders, as previously mentioned, the display order is not found in the Web page. Furthermore, claim 19 specifically recites that the queue "compris[es] a list of said supplemental files in said order". Anders does not meet this limitation. In Anders, there is no list of the file order. The order is given by the manner in which the data itself is interleaved in the packet. This is an entirely different concept.

Accordingly, claim 19 patentably distinguishes over Anders for many of the same reasons discussed above in connection with the other independent claims plus additional reasons.

Claims 20-22 depend from claim 19 and, therefore, distinguished over Anders for at least all of the reasons set forth above in connection with claim 19.

In addition, dependent claims 21 and 22 recite essentially the same subject matter as previously discussed in connection with dependent claims 7 and 8, respectively. Accordingly, they even further distinguish over the prior art for the same reasons given above in connection with dependent claims 6 and 7.

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: 4.25.05

Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

Attorney for Applicant